Dynamic Active Networks Services (DANS) as dynamic distributed systems

Carlo Tarantola,
Rajmund Paczkowski
Oracle Corporation
Mobile and Wireless Center of Expertise,
Warsaw, Poland
carlo.tarantola@oracle.com ,
rajmund.paczkowski@oracle.com

ABSTRACT

We highlight research done at Oracle's Warsaw Mobile and Wireless Center of Expertise to develop and manage in a dynamic way new services for telecommunications/data networks. Called Dynamic Active Networks Services (DANS), the architecture combines legacy and new components of a network with a set of protocols and algorithms by which networks operators and service providers can actively provide users with a wide range of services. We propose architecture and protocols for defining new services. The architecture treats the entire service environment as a dynamic, distributed system and we leverage concepts from Active Networking to deal with active nodes providing services within a network. The described framework is a different view of distributed systems: the novelty is the ties between distributed systems and active networking.

Keywords: Active Networks, Distributed Systems, Mobile Communications

INTRODUCTION

After the paper published as the cover story of from Isenberg [1] a big debate was brought into light regarding the impact of packet-centric IP technologies and the traditional circuit-switched-centric voice telecommunication approach. At the time, the extremes of the discussion were about "all intelligence at the core of the network" and "all the intelligence distributed at the edge". In that context the "network" is seen as a transport engine of multimedia data to support intelligent plug and play of components. Additionally, the discussion had two extremes: the "core networks" and the "edges of the network".

At the same time and in parallel to this, research in Active Networking, networks you can add programs to and execute some forms of customization to define specific applications, moved from the first discussions in the DARPA[1] in 1994-1995, to a more mature stage: projects like ANTS at the MIT [2], Netscript at Columbia [3], the Active Network Backbone (ABONE) at ISI and SRI [4] and others, evidenced the big potential of concentrating not just on the two extremes of the networks, but on the network as a whole "active" system and environment. The vision of these active packets (capsules) circulating in the network and embedding miniature programs and user data opens the door to a even more advanced concept that question what in fact "the" network is.

The additional component of our reflection is given by the current trend in mobile communications: millions of sophisticated devices such as PDAs[2], Smartphones, Laptops, Tablet PCs, are available and inter-connected thanks to network connectivity communication mechanisms, such as WiFi [5], GPRS3 [6] and one day maybe UMTS [7], that these devices do have built in. In addition to the above, an increasing interest has been put by the automotive industry into the world of Telematics, which refers to the provision of two-way voice and data communication between a vehicle and the Information Service Providers (ISPs).

This tremendous amount of wireless interconnectivity is giving "user" mobility an unprecedented importance and has precipitated the need for the ubiquitous availability of "a" network in order to enable the ability to share and find up-to-date information and bring a set of feature rich services to that user. This shift or evolution points to a new era of Mobile Tele-computing where Services On Demand or Value Added Services are at the heart.

---

[1] Defense Advanced Research Project Agency

[2] Personal Digital Assistant such as HP iPAQ, Palm Computing, or one of the Linux powered devices like the Sharp Zaurus

[3] General Packet Radio Service

PROBLEM STATEMENT

Many people work with several different (electronic) devices. A businessman may well have a laptop, a personal organizer and/or a mobile phone. Moreover printers, scanners, VCRs, cameras, DVD players, CD players and a myriad of other very used devices belong to our everyday life. What people want to do with all these machines is to be able to use them and benefit of the services provided by the networks in a seamless way, without even knowing the existence of such services in a specific place.

The DANS Framework is an architecture that provides the possibility to compose such devices and services they provide, into a single, dynamic distributed system. The resulting federation provides simplicity of access, facility of administration, and support in shared information, which is provided by a large monolithic system. It also retains the flexibility and control provided by a personal computer or workstation.

The most important concept within this architecture is that of service. Services can be divided into pull-based and push-based. In the former case, services are activated on user demand, e.g. server keeps latest news in categories and user selects the appropriate category from which he wants to download something. In the latter, user's subscription drives server behavior and, referring to the user's profile, server pushes information when triggered by some event, e.g. arrival of a new mail or an update for a stock quote.

An active framework service is not only some specific information required by the user, such as stock rates or train time schedules, but is also a reactive mechanism that would enable activities such as for the user to send this information to the nearest printer, or to send an email/attached image to any new device that has just been connected to the network, without the help of a system administrator. The consequence is that the service provider may surely be a software house, which operates the service for the stock exchange, but also the electronic device that is empowered with the task of doing something.

As a consequence, our problem is to find a simple way to plug in and manage services and devices, into a networked environment, giving instant access to an array of services than the user is not even subscribed to.

We are not going to worry of the billing aspect for these services because billing for these kind of micro payments can nowadays be made very simple, secure and immediate thanks to services as Premium SMS (PSMS). PSMS is a solution already in place and already widely in use in Europe as a transactional mechanism to buy ring tones and other downloads by sending an SMS to a special number. The basic idea is that, while a “standard” SMS is similar to a local phone call, a PSMS is more like a "special" number, with per-message fees that could total several euros or more. Basically when a PSMS is sent, the telephone operators see a “billing event” generated and, as a consequence, they append a special charge to the sender's monthly phone bill.

DESCRIPTION

DANS (Dynamic Active Network Services) [8] framework is a set of software components, which supports creating peer-to-peer networking in heterogonous environments. DANS is based on SIP to locate, register and use the services and mobile code that is available in the network. In this sense we note that a processing language and mobile code can be considered examples of active networks at the application layer, in fact we have Java servlets that receive call from the SIP server (handling the logic), while the mobile code carries in the request itself and instructs the SIP on how to handle the requests. The Session Initiation Protocol (SIP) provides advanced signaling and control functionality for a wide variety of multimedia services [9]. SIP is used as an application-layer control protocol for initiating, modifying, and terminating sessions with one or more participants [10]. The content of a session may be of an arbitrary type, and users participating in the session may communicate via multicast or unicast relations, or a combination of these.

DANS provides:

- Service oriented environment which enables to provide services by every kind of device connected to Internet
- Simplicity and low cost of service creation and delivery
- Service independent architecture
- Easy ways of integration with SIP (Session Initiation Protocol) environments.
- Effective and convenient access to essential services

Additionally, a simple reference network was created to show how to use the framework and how to build services running on it.

Main idea of DANS is to provide dynamic network where various devices are connected and they are able to provide services to each other. Authors of this solution focused on the possibility of using mobile devices as equivalent “partners” in providing and using services in collaboration with other mobile devices as well as standalone computers.

The ideal situation would be when each device is to connect to another device and use its services directly. Unfortunately there are no simple ways to make direct data connections between mobile phones or between mobile phones and standalone computers. On the other hand almost every kind of communication device has the possibility of connecting to the Internet – e.g. via GPRS gateways (mobile phones) or simple network connections (standalone computers).

All these facts put together pushed us to create the Mediator application. This application, called P2P Gateway is the place where all devices participating in exchanging data meet each other. Additionally, on each device connected to DANS networks we run an application called P2P Host, which is responsible for managing services and connections with the rest of network. P2P Hosts implementations vary with the device kind, however, thanks to J2ME technology, which is accessible on most mobile and standalone platforms, it was possible to create common set of classes, common to all kinds of Java

enabled devices. This set of classes is one of integral part of the DANS framework and its called P2P Host Stub. It has the functionality, which enables to provide and use services in an easy to use, platform independent way. To provide user and service directories, P2P Name/Service Registry was created. It is an application part, which consists of database schema keeping all necessary data to proper function of the system and the set of classes managing this data.

In the following sections of the paper we describe all these components as well as the main architecture and collaboration rules.

COLLABORATION SCENARIOS

The DANS Framework supports peer-to-peer connections, but direct data exchanging is mediated by a central application: the P2P Gateway. As it is shown on the picture below, each device that is intended to participate in a DANS network has to connect to the gateway. A physical detail of the connection is dependent on the type of that particular device. In its basic version the P2P Gateway supports only socket connection. Unfortunately a large part of mobile devices, especially mobile phones, is able to perform only HTTP connections. To meet this need, a special kind of HTTP servlet has been built and is provided. The basic functionality of this servlet is to emulate socket connections based on HTTP connections in combination with HTTP sessions. The servlet, if needed, is deployed on any application server supporting Java Servlet technology.

Each operation, which is related to creating peer-to-peer link between devices, calling remote service or providing a service, is connected with the P2P Name/Service Registry. This is a place, where all user and service data are stored. Users are identified by email-like addresses, which may correspond to SIP addresses in other networks (e.g. instant messaging). We assume that each user can have more then one device connected to a network. In N/S Registry all services provided by all user's devices are visible as services of that particular user. There is no special distinction between different kind of devices of given user from system's point of view. Its possible to place "type of device" in the service description document that is assigned to each service, but it has only informative role for potential remote user.

Basic scenario of connecting to DANS network is:

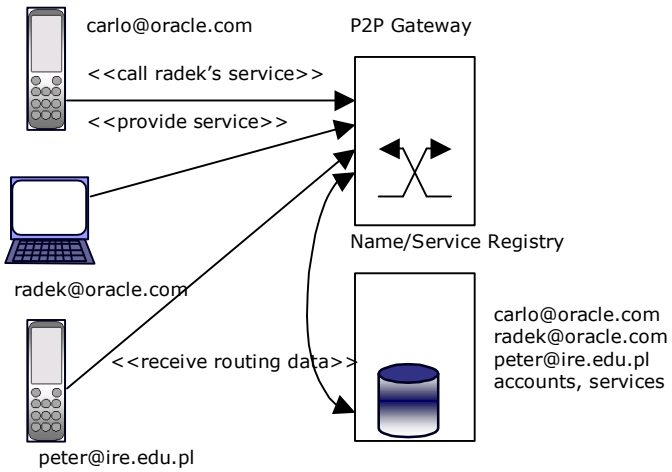
- P2P Host is connecting to P2P Gateway – after that P2P Host is registered in N/S registry but it has no owner, hence its invisible for rest of users (there's no anonymous devices)
- P2P Host is registering as a device of particular user (e.g. carlo@oracle.com) - after this operation device gains a owner which makes it visible as one of devices of this user, it can call services of other users but it can not provide services itself
- P2P Host is registering services which its intended to provide – after this step N/S Registry is informed about new services provided by user formerly registered; from this moment each participant of DANS network can see and call these services

When a device is connected to the DANS network it can use services provided by other network participants:

- P2P Hosts is sending to gateway request for service description of given user (e.g. radek@oracle.com)
- P2P Gateway is retrieving service descriptions associated with given user and sending it back to device which sent a query
- P2P Host is rendering proper user interface to use services (and actions within them)
- user of P2P Host can choose particular service and action within it, place a set of needed parameters and execute the action
- P2P Host is waiting for a execution result and if it comes rendering in user interface (waiting for a response is not blocking operation; in the meantime any other operations could be performed)

Typical service invocation runs in a request-response schema, but there is a possibility of using notifications as an asynchronous way of responding. One device can e.g. subscribe itself as incoming mail notification receiver.

There is a simplified XML-based protocol, which enables the devices to perform all the operations we mentioned. This protocol is, in its base, very similar to web services protocols such as XML-RPC or SOAP. On one hand it contains many extensions dedicated to peer-to-peer networking but on the other hand it has many restrictions when compared to web services protocols, the reason being the limited computational resources of mobile devices. This native protocol is able to describe all services and actions within them as well as a way of rendering user interface for it. Thanks to this fact it was possible to create a universal client for all kind of services. On the other hand each service has to implement special kind of interface and has to be placed in special place. These requirements are caused by dynamic service invoker, which manages all provided services and service calling engine. P2P Gateway connection architecture is shown below:



Agent can talk to agent only when P2P Session object is created. Even when device is connecting to gateway and beginning registration process, its agent sends data to another agent – system agent. It is the only agent that exists during whole time when gateway is running. Registering device in N/S Registry is performed in similar way like invoking service on other device. P2P has its own P2P Host Stub, which provides services related to registering, unregistering etc. Representing devices by agents, which are device independent, enables connections and data exchanging between any types of devices, connected to gateway.

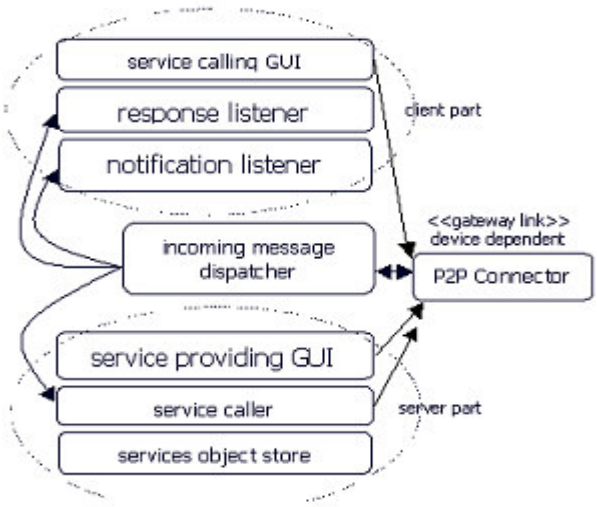
There is a reference implementation of the P2P Gateway. It was built on Java 2 Standard Edition as a standalone application ready to run on Windows and Linux/Unix platforms.

P2P HOST HUB

P2P Host Stub is the common part for all P2P Hosts connected to DANS network. It consists of a set of classes and interfaces defining basic functionalities, and is described in the picture below. P2P Host is divided in two parts: the server part and client part.

The Server part is responsible for managing services. Here all the service classes are stored and all initialization and invocation processes are performed. On the top level of this part exists the user interface, which is strongly related to the specific type of device. P2P Host Stub defines only the set of interfaces, which should be implemented in dependency on device kind.

Client side consists of response and notification collectors, which intercept proper messages from gateway, and perform operation dependent of device and platform type. On the top of this part resides the GUI part tied with service calling engine.



P2P Host Stub Schema

There are many types of messages received from P2P Gateway. Message dispatcher differentiates these messages on the basis of the incoming XML document types.

One of the most device dependent modules in the P2P Host is the P2P Connector. It is a set of interfaces, which covers physical details of connection e.g. end device programmer has to implement method sendMessage(String str) to describe how a particular device will send a message to the gateway.

At the moment there are two reference P2P Host implementations. One is the P2P Host running on standalone computers with Java 2 Standard Edition. The second one is a Java 2 Micro Edition client implemented as a midlet which can be deployed on any mobile device supporting J2ME MIDP 1.0 or MIDP 2.0.

EXAMPLE

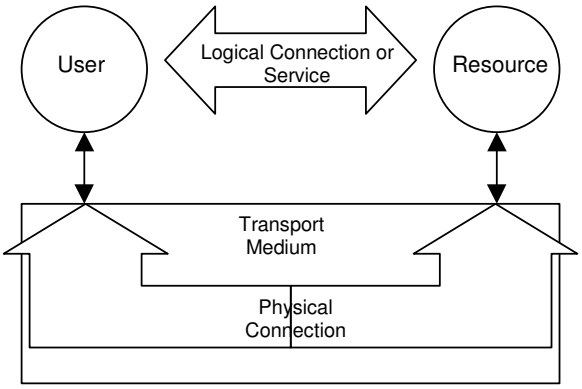
We have a fleet of smartphones and WLAN enabled PDAs. One of these phones is configured as the “fleet manager” (FM). The SIP URI of a FM REGISTER which SIP URIs it is responsible for. For example this could be the case of a family where the father and the mother wants to supervise the telephone spending of their teenager children.

The scenario shows a context sensitive situation where devices can be brought into usage while the user enters into a specific geographical area (e.g. a mall). This usage requires the payment of a small fee that can be authorized by the fleet manager. The active nodes in the network can offer services such as keeping track of the total cost for some small amounts of money paid during the session (defined as the period of time between the moment you enter the geographical area an when you leave). These charges represent the costs of the different services used by the people that belong to the fleet. Not only these amounts are presented to the FM as a PSMS for authorization, but also a total is indicated. Typical case is to authorize (or not) children to pay for some amusements or candies while they are in a mall. Parents don’t need to provide with cash or credit cards as the transactions are done via PSMS and parents don’t need to be physically located in the same geographical area. [Details].

Note that this represents also a clean way of dealing with push services because the PSMS is generated only when an authorization is required.

As remarks note that the above situation destroys the classical boundaries of Intranet and Extranet. There are two major implications to it:

- The first implication is that some transport medium can be used and programmed to enable user to access resources and, resources, to provide content on a per-request basis.



- The second implication is that the services will be implemented as an application software layer sitting on top of the transport medium. This software layer is going to be responsible for all the communications, requests and services creation that can be thought of. Especially software and its integration with the physical layer will be of paramount importance to enable end-to-end management, security, quality of services and "virtuality".

The trend is to have something as simple as a plug in the wall or a cheap wireless connection that links all kinds of devices to a wide range of services. Moreover services themselves will always be available to save time and money, and safeguard privacy.

CONCLUSIONS

We presented our work to build applications that benefit of active nodes in a network. These active nodes offer services that are triggered and potentially created by the same applications. Thanks to the support of location aware mechanisms, the result is that true context sensitive, presence ready applications can be built and deployed. Moreover we are able to have (Active) Networks, which identify users and content in order to optimize delivery according to business priorities.

The described framework is a different view of distributed systems. By means of it it's possible to create dynamic active networks, which give:

- Capabilities of providing services in very simple way (only by putting service objects in service store of our P2P Host)
- Equal chance for mobile and standalone devices to call and provide services (relative to their resources)
- Capabilities of creating grid architecture which would enable to take advantage of big processing resources sleeping in computer idle time
- Very low cost of implementation (depending on scale, but low in general)
- Integration with SIP compliant applications – e.g. instant messengers)
- Flexible architecture which allows to create new extensions to existing P2P networks

Further plans of extending DANS Framework are related to distribution of device agents and gateways. This will enable global P2P Network with almost infinite scalability. No less important is issue of security and access control. All these problems are under consideration and it's a matter of short time, when new DANS Framework and new reference implementation will arise.

REFERENCES

1. David S. Isenberg, *The Dawn of the "Stupid Network*", ACM Networker 2.1, February/March 1998, pp. 24-31
2. David J. Wetherall, John Guttag, and David L. Tennenhouse, *ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols*, IEEE OPENARCH'98, San Francisco, CA, April 1998
3. Y. Yemini and S. da Silva, *Towards Programmable Networks*, IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, L'Aquila, Italy, October, 1996.
4. Steven Berson, Bob Braden, Livio Ricciulli, *Introduction to the Abone*, June 15, 2000, http://www.isi.edu/abone/DOCUMENTS/ABoneIntro.pdf
5. *WiFi Standards*: http://standards.ieee.org/getieee802/
6. *What is General Packet Radio Service?*, http://www.gsmworld.com/technology/gprs/intro.shtml
7. http://www.umts-forum.org/
8. Carlo Tarantola, *Dynamic Active Network Services*, Fifth International Conference on Mobile Data Management, 19 - 22 January 2004, Berkeley, California.
9. Schulzrinne, Rosenberg, *The Session Inititation Protocol: Internet Centric Signaling*, IEEE Communications Magazine, October 2000, pp. 134-141
10. Handley, Schulzrinne, Schooler and Rosenberg: *RFC 2543 SIP: Session Initiation Protocol*. November 2000